Question 1

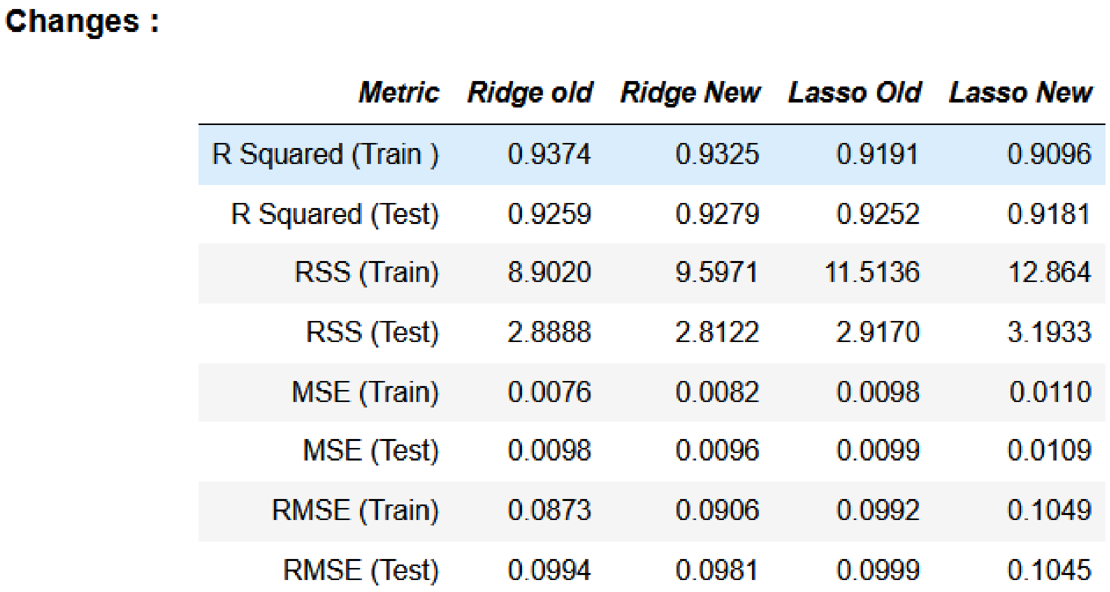
What is the optimal value of alpha for ridge and lasso regression? What will be the changes in the model if you choose double the value of alpha for both ridge and lasso? What will be the most important predictor variables after the change is implemented?

Answer :

Optimal value of alpha for Ridge Regression = 10
Optimal value of alpha for Lambda Regression = 0.001

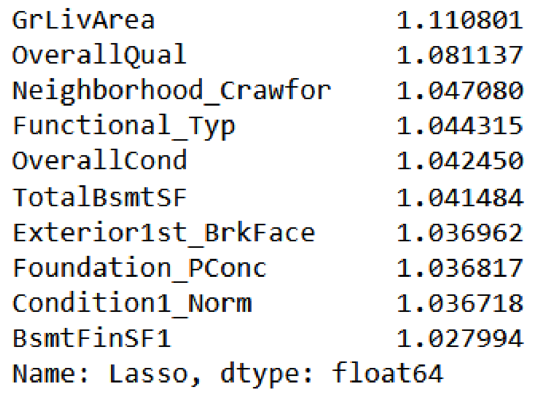
Doubling the alpha values would result in the alpha for Ridge Regression to be 20 and Alpha for Lambda Regression to be 0.002. For reference lets name thes as “New” and the original as “Old”

In the code I added the changes and evaluated the models by printing the R2, RSS, MSE and RMSE values for both Train and test datas and are as follows:

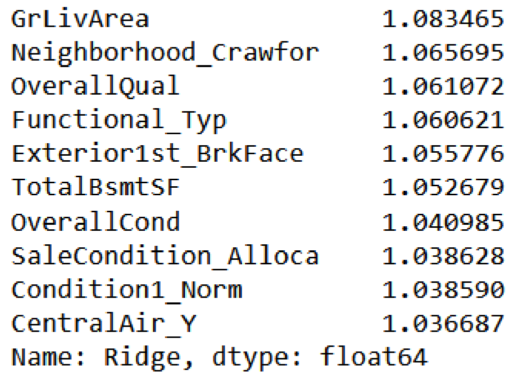


Changes :

Metric	*Ridge old*	*Ridge New*	*Lasso Old*	*Lasso New*
R Squared (Train)	0.9374	0.9325	0.9191	0.9096
R Squared (Test)	0.9259	0.9279	0.9252	0.9181
RSS (Train)	8.9020	9.5971	11.5136	12.864
RSS (Test)	2.8888	2.8122	2.9170	3.1933
MSE (Train)	0.0076	0.0082	0.0098	0.0110
MSE (Test)	0.0098	0.0096	0.0099	0.0109
RMSE (Train)	0.0873	0.0906	0.0992	0.1049
RMSE (Test)	0.0994	0.0981	0.0999	0.1045



```
GrLivArea                1.110801
OverallQual              1.081137
Neighborhood_Crawfor     1.047080
Functional_Typ           1.044315
OverallCond              1.042450
TotalBsmtSF              1.041484
Exterior1st_BrkFace      1.036962
Foundation_PConc         1.036817
Condition1_Norm          1.036718
BsmtFinSF1               1.027994
Name: Lasso, dtype: float64
```



```
GrLivArea               1.083465
Neighborhood_Crawfor    1.065695
OverallQual             1.061072
Functional_Typ          1.060621
Exterior1st_BrkFace     1.055776
TotalBsmtSF             1.052679
OverallCond             1.040985
SaleCondition_Alloca    1.038628
Condition1_Norm         1.038590
CentralAir_Y            1.036687
Name: Ridge, dtype: float64
```

From the above it appears that the most important predictor variables are GrLivArea, OverallQual, Neighborhood_Crawfor, Functional_Typ, OverallCond, TotalBsmtSF,Exterior1st_BrkFace, Foundation_PConc, Condition1_Norm , BsmtFinSF1¶

Question 2

You have determined the optimal value of lambda for ridge and lasso regression during the assignment. Now, which one will you choose to apply and why?

Answers

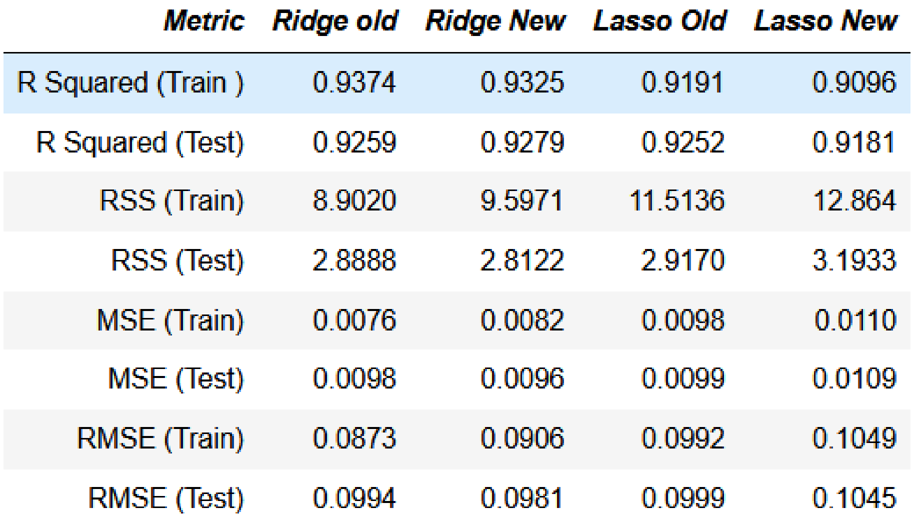
The choice between Ridge and Lasso regression, after determining the optimal value of lambda depends on the specific problem and the characteristics of your dataset. Additionally Lasso tends to yield more interpretable models by setting some coefficients to zero, which can be valuable in some applications.

However, Ridge is computationally less intensive than Lasso when you have a large number of features. If computational efficiency is a concern, this might influence the choice.

If Lasso has a higher R-squared and lower MSE, it may be the better choice, especially if feature selection is important.

If Ridge has a higher R-squared and lower MSE, it may be the better choice, especially if multicollinearity is a concern.

Changes :



Metric	Ridge old	Ridge New	Lasso Old	Lasso New
R Squared (Train)	0.9374	0.9325	0.9191	0.9096
R Squared (Test)	0.9259	0.9279	0.9252	0.9181
RSS (Train)	8.9020	9.5971	11.5136	12.864
RSS (Test)	2.8888	2.8122	2.9170	3.1933
MSE (Train)	0.0076	0.0082	0.0098	0.0110
MSE (Test)	0.0098	0.0096	0.0099	0.0109
RMSE (Train)	0.0873	0.0906	0.0992	0.1049
RMSE (Test)	0.0994	0.0981	0.0999	0.1045

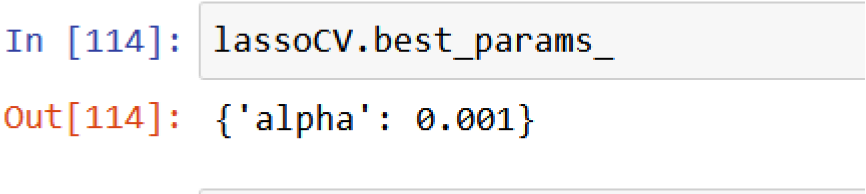
Comparing the table, R-squared for Ridge is higher, and MSE is Lower. So it might be best to Pick Ridge Here

Question 3

After building the model, you realised that the five most important predictor variables in the lasso model are not available in the incoming data. You will now have to create another model excluding the five most important predictor variables. Which are the five most important predictor variables now?

Answer :

['GrLivArea','Neighborhood_Crawfor', 'Exterior1st_BrkFace', 'OverallQual', 'Functional_Typ'] are the top 5 predictor variables.

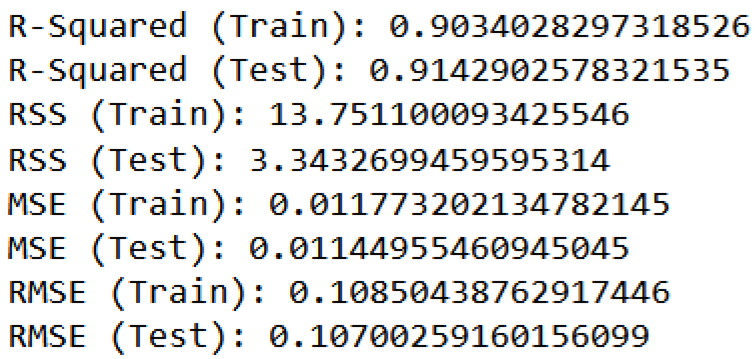


```
In [114]: lassoCV.best_params_

Out[114]: {'alpha': 0.001}
```

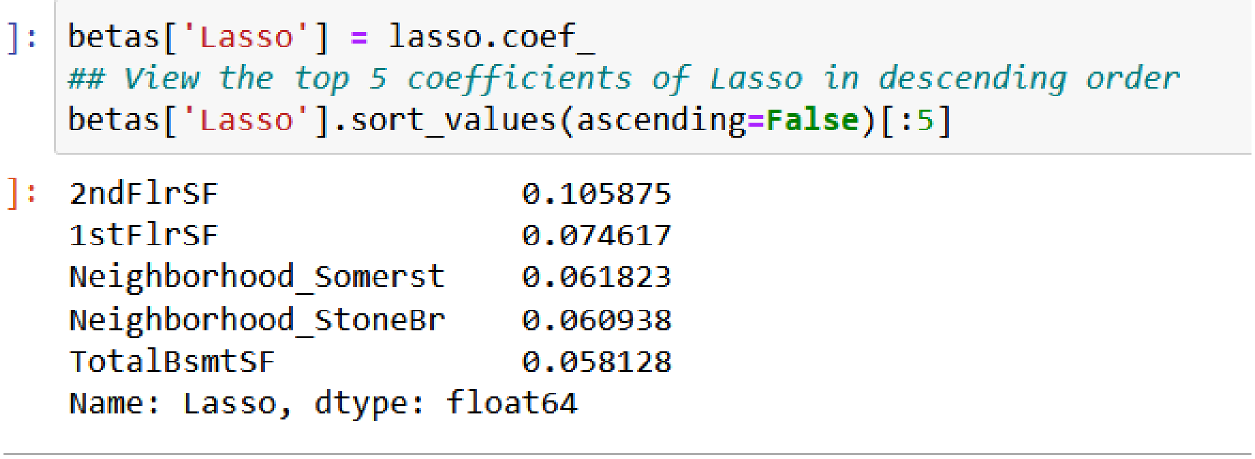
On removing them : The new alpha value is 0.001

With the new model, the evaluation is as follows :



```
R-Squared (Train): 0.9034028297318526
R-Squared (Test): 0.9142902578321535
RSS (Train): 13.751100093425546
RSS (Test): 3.3432699459595314
MSE (Train): 0.011773202134782145
MSE (Test): 0.011449554609450445
RMSE (Train): 0.10850438762917446
RMSE (Test): 0.10700259160156099
```

And fetching the top 5 coefficients



```
]: betas['Lasso'] = lasso.coef_
   ## View the top 5 coefficients of Lasso in descending order
   betas['Lasso'].sort_values(ascending=False)[:5]
```

```
]: 2ndFlrSF                0.105875
   1stFlrSF                0.074617
   Neighborhood_Somerst    0.061823
   Neighborhood_StoneBr    0.060938
   TotalBsmtSF             0.058128
   Name: Lasso, dtype: float64
```

Hence the top 5 predictor variables after dropping the top 5 from the previous model are :

- 2ndFlrSF
- 1stFlrSF
- Neighborhood_Somerst
- Neighborhood_StoneBr
- TotalBsmtSF

Question 4

How can you make sure that a model is robust and generalisable? What are the implications of the same for the accuracy of the model and why?

Answers

To ensure a model is robust and generalizable:

- Use Sufficient Data: Collect a diverse and representative dataset with an adequate sample size to capture underlying patterns.

- Cross-Validation: Employ techniques like k-fold cross-validation to assess how well the model generalizes to new data, reducing overfitting.
- Feature Engineering: Carefully preprocess and select features, removing noise and irrelevant information.
- Regularization: Apply regularization techniques like Ridge and Lasso to prevent overfitting and promote model stability.
- Hyperparameter Tuning: Optimize model hyperparameters via techniques like grid search or random search.
- Ensemble Learning: Combine multiple models to improve generalization and reduce variance.

Implications for Accuracy:

- Robustness and generalizability often come at a slight cost to training accuracy, as models become less sensitive to noise in the training data.
- The aim is to strike a balance between model bias and variance. A highly accurate model on the training data may overfit and perform poorly on unseen data.
- A robust, generalizable model maintains its performance across different datasets and real-world scenarios, enhancing reliability in practical applications.
- Prioritizing accuracy on training data at the expense of generalizability can lead to models that fail to perform well in the real world, limiting their utility.